USEFUL FOR UNIVERSITY EXAMS, GATE, NET AND OTHER CS EXAMS

DATABASE MANAGEMENT SYSTEM

DATABASE LANGUAGES

PROPER NOTES IN PPT FORM

PART -5



DATABASE LANGUAGES

- Languages in general help us to communicate and share our thougts with other people.
- It helps us to convey our thoughts to other people.
- Similarly, DATABASE languages help the users to communicate with the databases.
- DBMS uses queries to access the database.
- Database languages are used to perform a variety of critical tasks that help a database management system function correctly.
- Database languages help to read, store and update the data in the database.

TYPES OF SQL COMMANDS

Types of SQL Commands			
DDL	DML	DCL	TCL
CREATE ALTER DROP TRUNCATE RENAME	SELECT INSERT UPDATE DELETE MERGE	GRANT REVOKE	COMMIT ROLLBACK SAVEPOINT

DATA DEFINITION LANGUAGE

- DDL is used to specify the database schema.
- It is used to specify the structure of a database.
- We can alter the structure of database or drop a complete database from the memory of computer.
- DDL can be used to create tables, schemas, indexes, constraints in a database.
- Since DDL commands change the structure of table and not the data therefore these are mainly used by DBA and not the end user.

CREATE COMMAND

- DDL consists of following commands:

1. **CREATE** – It is used to create objects in the database.

To Create a Database the syntax is -

CREATE DATABASE database_name;

For eg : CREATE DATABASE emp;

To Create a Table the syntax is:

```
CREATE TABLE table_name(
 Column_name datataype(size) constraint,
 Column_name datataype(size) constraint );
```

For eg:

```
CREATE TABLE Persons (
    PersonID int,
    LastName varchar(255),
    FirstName varchar(255),
    Address varchar(255),
    City varchar(255));
```

ALTER COMMAND

2. **Alter** – It is used to modify the structure of database.

Syntax To add New Column in DB:

Alter Table table_name ADD(

Column_1 DATATYPE,

Column_2 DATATYPE,

Column_3 DATATYPE);

Eg: Alter Table emp ADD(

Salary integer(10));

Syntax To change Datatype of a Column:

Alter Table table_name

MODIFY column_name DATATYPE;

Eg: Alter Table emp

MODIFY Salary float;

ALTER COMMAND

Syntax to Remove a Column from the table:

ALTER TABLE table_name

DROP COLUMN column_name;

Eg: ALTER TABLE emp

DROP COLUMN Salary;

Syntax to rename a column a column in a Table:

ALTER TABLE table_name

RENAME COLUMN old_column_name TO new_column_name;

Eg: ALTER TABLE emp

RENAME COLUMN Employee_Name TO E_name;

DROP & TRUNCATE COMMAND

3. **Drop** – It is used to drop the database object completely from the memory.

To DROP a DATABASE :

DROP DATABASE database_name;

Eg: DROP DATABASE emp;

To DROP a TABLE:

DROP TABLE table_name;

Eg: DROP TABLE student;

4. **Truncate** – It is used to remove all the records from the table. Although TRUNCATE TABLE is similar to DELETE , it is classified as a DDL statement rather than a DML statement.

It differs from DELETE in the following ways: Truncate operations drop and re-create the table, which is much faster than deleting rows one by one, particularly for large tables.

It does deallocates the space from the memory of the data which was present earlier in the table.

Syntax to Truncate the table:

TRUNCATE TABLE table_name;

Eg: TRUNCATE TABLE emp;

RENAME COMMAND

5. **Rename** – It is used to rename an object.

Syntax to RENAME a TABLE :

RENAME old_table_name TO new_table_name;

Eg: RENAME Salary To Total_Salary;

Syntax to RENAME a COLUMN of a TABLE:

ALTER TABLE table_name

RENAME COLUMN old_column_name TO new_column_name;

All of these commands effect the schema of database and hence come in DDL category.

DML LANGUAGE

- A Data Manipulation Language(DML) is a language that enables users to access or manipulate data of the database.
- It handles user requests as they are responsible for data modification.
- These commands are used to perform certain operations such as insertion, deletion, updation, and retrieval of the data from the database.
- These commands can be roll backed.

SELECT COMMAND

Select: It is used to retrieve data from a database.

To SELECT data From the entire Database:

SYNTAX : SELECT * FROM Table_Name;

Eg: SELECT * FROM stu;

S_ID	S_Name	S_Add	S_Marks
1	A	X	30
2	B	Y	32
3	C	Z	35

To SELECT specific columns from a Table:

SYNTAX: SELECT col1, col2, col3 FROM Table_Name;

Eg: SELECT S_ID, S_Name FROM stu;

S_ID	S_Name
1	A
2	B
3	C

SELECT COMMAND

To SELECT specific columns and specific rows from a Table:

SYNTAX: SELECT col1, col2, col3 FROM Table_Name

WHERE condition;

Eg: SELECT S_ID, S_Name FROM stu

WHERE S_Marks>30;

S_ID	S_Name	S_Add	S_Marks
1	A	X	30
2	B	Y	32
3	C	Z	35

S_ID	S_Name	S_Add	S_Marks
2	B	Y	32
3	C	Z	35

INSERT COMMAND

2. Insert: It is used to insert data into a table.

To INSERT data in ALL columns:

SYNTAX: INSERT INTO(col1,col2,col3,col4) VALUES(val1,val2,val3,val4);

Eg: `INSERT INTO Customers(CustomerName, ContactNum, Address, City, PostalCode, Country) VALUES("Preeti", 1234567890, "H.No.21 A.S. Colony", "Karnal",123456, "India");`

Other way :

Eg: `INSERT INTO Customers VALUES("Preeti", 1234567890, "H.No.21 A.S. Colony", "Karnal",123456, "India");`

To INSERT Incomplete data :

Eg: `INSERT INTO Customers(CustomerName, ContactNum) VALUES("Ritika", 1245678944);`

UPDATE COMMAND

3. Update: It is used to update existing data within a table.

SYNTAX: UPDATE table_name

SET column1 = value1,

SET column2 = value2,

SET column3 = value3

WHERE condition;

S_id	S_name	S_phone
1	Preeti	1234
2	Ritik	12345

Eg:
```
UPDATE stu
SET s_name="Deepak", s_phone=123456
WHERE s_id=2;
```

S_id	S_name	S_phone
1	Preeti	1234
2	Deepak	123456

Here, the SET statement is used to set new values to the particular column, WHERE clause is used to select rows for which the columns are updated for the given table.

DELETE COMMAND

- **DELETE :** It is used to delete existing records from a table, i.e., it is used to remove one or more rows from a table.

To delete rows from a table based on a condition :

SYNTAX: DELETE FROM table_name

WHERE condition;

Eg.: DELETE FROM stu

WHERE stu_id=3;

To delete all rows from a table/ complete data of table:

SYNTAX: DELETE FROM table_name;

Eg: DELETE FROM stu;

S_Id	S_name	S_addr
1	Preeti	Pnp
2	Ritika	Delhi
3	Nitika	Sonipat

S_Id	S_name	S_addr
1	Preeti	Pnp
2	Ritika	Delhi

S_Id	S_name	S_addr

DELETE/ DROP DIFFERENCE

- The DELETE statement only removes the data from the table, whereas the TRUNCATE statement also frees the memory along with data removal. Hence, TRUNCATE is more efficient in removing all the data from a table.

TYPE OF DML LANGUAGE

- **PROCEDURAL DML (NON DECLARATIVE)** – They require a user to specify what data are needed and how to get those data. It requires the user to specify the steps that the system should take to manipulate the data. Examples of procedural DMLs include languages such as COBOL, FORTRAN, and PL/SQL.
- **NON PROCEDURAL DML (DECLARATIVE)** – They require a user to specify what data are needed without specifying how to get those data. Non -Procedural DMLs are usually easier to learn and use than Procedural DMLs. For eg: SQL, Oracle, MySQL

DATA CONTROL LANGUAGE

- It is used to control access to the stored or saved data in database.
- It is used to manage the privileges that can be given to the database users.
- DCL can be used to access this stored data, and it is used mainly to revoke and grant the users the required access to any database.
- We require data access permissions to execute any command or query in the database system. This user access is controlled using the DCL statements.
- DCL commands are transactional i.e., these commands include rollback parameters.
- It consists of following commands :

GRANT & REVOKE

Grant: It is used to give user access privileges to a database objects.

SYNTAX: GRANT Privilege

ON Object

To user1, user2;

Example: **GRANT** ALL **ON** mystudentdb.* **TO** john@localhost;

Revoke: It is used to take back permissions from the user.

SYNTAX: REVOKE Privilege

ON Object

To user1, user2;

Example: **GRANT** read **ON** mystudentdb.* **TO** john@localhost;

TRANSACTION CONTROL LANGUAGE

- Transaction Control Language (TCL) is a set of special commands that deal with the transactions within the database.
- A transaction is a collection of related tasks that are treated as a single execution unit by the DBMS software.
- Hence, transactions are responsible for the execution of different tasks within a database.
- The modifications performed using the DML commands are executed or rollbacked with the help of TCL commands.
- These commands are used to keep a check on other commands and their effects on the database.

COMMIT

COMMIT :

- It is used to permanently save all the modifications are done (all the transactions) by the DML commands in the database. Once issued, it cannot be undone.
- DBMS software implicitly uses the COMMIT command before and after every DDL command to save the change permanently in the database.
- SYNTAX:

  COMMIT;

ROLLBACK

ROLLBACK :

- It is used to undo the transactions that have not already been permanently saved (or committed) to the database.
- It restores the previously stored value, i.e., the data present before the execution of the transactions.
- SYNTAX:

ROLLBACK;

To undo a group of transactions to a certain point :

ROLLBACK TO savepoint_name;

SAVEPOINT

- **SAVEPOINT :**
- It is used to create a point within the groups of transactions to save or roll back later.
- It is used to roll the transactions back to a certain point without the need to roll back the whole group of transactions.
- SYNTAX:

```
SAVEPOINT savepoint_name;
```

To release a savepoint :

```
RELEASE SAVEPOINT savepoint_name;
```

THANK YOU

